No artigo, Parnas critica a prática comum da época de decompor sistemas em módulos com base no fluxo de processamento (etapas sequenciais, como entrada, processamento e saída). Ele argumenta que esse critério torna os sistemas difíceis de manter e evoluir.

O maior progresso na área de programação modular tem sido o desenvolvimento de técnicas de codificação e desenvolvedores que permitem que um módulo seja escrito com pouco conhecimento do código usado em outro módulo e que módulos sejam refeitos e substituídos sem a refatoração de todo o sistema.

Na prática comum, a modularização era feita com base no fluxo de processamento, ou seja, o sistema era dividido em etapas que correspondem à sequência de operações (entrada, transformação, saída). Essa forma pode ser considerada como intuitiva, mas Parnas mostra que ela leva a sistemas com desvantagens do tipo: Difíceis de modificar, pois uma alteração em uma parte frequentemente exige mudanças em várias outras; Pouco flexíveis, já que novas funcionalidades não se encaixam facilmente na estrutura rígida; Difíceis de compreender e testar, porque os módulos dependem fortemente uns dos outros.  
  
  
Parnas exemplifica sua ideia apresentando duas formas possíveis de modularizar um mesmo problema. A primeira, baseada em fluxo de processamento, resulta em módulos fortemente interdependentes e frágeis diante de mudanças. A segunda, orientada pela ocultação de informação, embora menos óbvia inicialmente, mostra-se superior, pois isola as áreas mais suscetíveis a mudanças e facilita tanto a evolução quanto a manutenção do sistema. Essa comparação evidencia que a qualidade de um design modular não está apenas em refletir a ordem lógica de execução, mas em prever e minimizar o custo de futuras modificações.

As vantagens da abordagem proposta pelo autor do artigo são muitas. A manutenção do sistema torna-se mais simples, pois alterações em decisões de projeto ficam restritas ao módulo em que essas decisões foram encapsuladas. Além disso, a divisão clara entre módulos permite que equipes de desenvolvimento trabalhem em paralelo sem depender do conhecimento dos detalhes internos de outros módulos. Outro ganho importante é a reusabilidade, já que módulos bem definidos podem ser aproveitados em diferentes contextos. A testabilidade também é favorecida, pois cada módulo pode ser verificado de forma independente, o que aumenta a confiabilidade do sistema como um todo.

O impacto dessa proposta foi profundo e duradouro. O conceito de *information hiding* inaugurou uma nova perspectiva sobre modularização, influenciando diretamente a programação estruturada, a orientação a objetos e, mais recentemente, arquiteturas baseadas em componentes e microsserviços. A grande contribuição de Parnas foi mostrar que a decomposição de sistemas deve priorizar não apenas o funcionamento imediato, mas sobretudo a capacidade de evolução futura com baixo custo.

Assim, a mensagem central do artigo é que a melhor maneira de estruturar sistemas em módulos não é seguir a sequência de processamento, mas sim projetá-los de forma a isolar as decisões de projeto suscetíveis a mudanças. Esse insight mudou a prática da Engenharia de Software, oferecendo bases sólidas para o desenvolvimento de sistemas mais robustos, adaptáveis e duradouros.